Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

Storage Management and Data Mining in High Energy Physics Applications

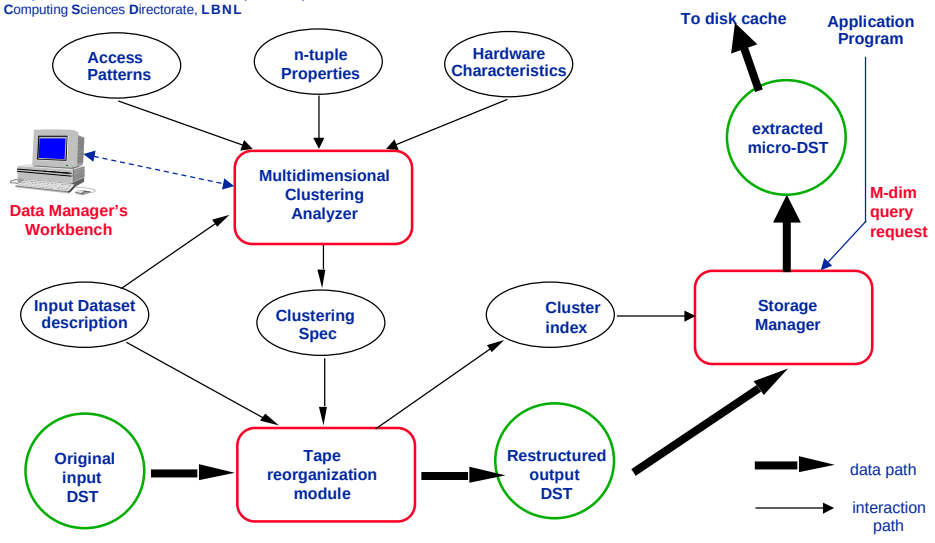
Arie Shoshani
Doron Rotem
Henrik Nordberg
Luis Bernardo
(http:/gizmo.lbl.gov/DM.html)

Scientific Data Management R&D Group
Lawrence Berkeley National Laboratory

October , 1997

Events Clustering and Access: Main Components

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL



Main Tasks

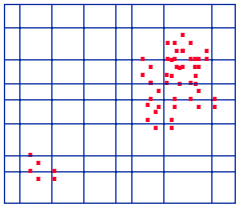
- Discover event clusters
  - based on natural distribution - Data Mining
  - based on access patterns - consult physicists
  - simulate performance - data manager's workbench
- Manage cluster access
  - given a query, determine clusters to access, use multi-dimensional indexes to select events that qualify
- Reorganize DST tapes according to clusters
  - long process - done initially, then rarely
  - flow control - restart after interruption
- Cache management
  - determine if in cache, which incremental clusters to cache, which clusters to purge from cache

Discover events clusters

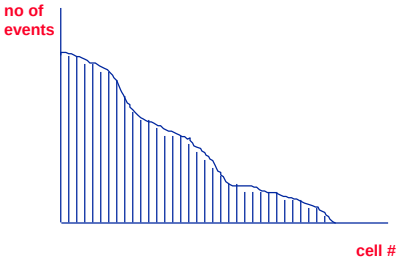
- Top down approach
  - partition each dimension into "bins" (e.g. 1-2 GEV, ..., 1-3 pions, ...)
  - select subset of dimensions based on physicist's experience
  - analyze which events fall into the same "cell" (i.e. m-dim rectangles formed by the bins)
  - eliminate empty cells
  - combine cells to form similar size clusters
- Assumption
  - most queries are "range queries"

Top Down Cell Management

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL



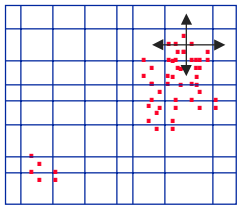
- **Assume:** 7 dimensions, 10 bins each
  - -- Number of cells: 10^7, 4 byte counters
  - -- **Number of bytes**: $4x10^7$, 40 MB
- **For e.g. small dataset 97% of cells are empty:**
  - -- store only populated cells
  - -- use hash tables to locate existing cells
  - -- use 2 bytes for bin_id per property: ratio for p% full is: 200/ (n+2)p
  - -- **No of bytes**: for 7 dim, 3% => 5.4 MB
- **Sort cells by size (number of events)**



Cluster discovery

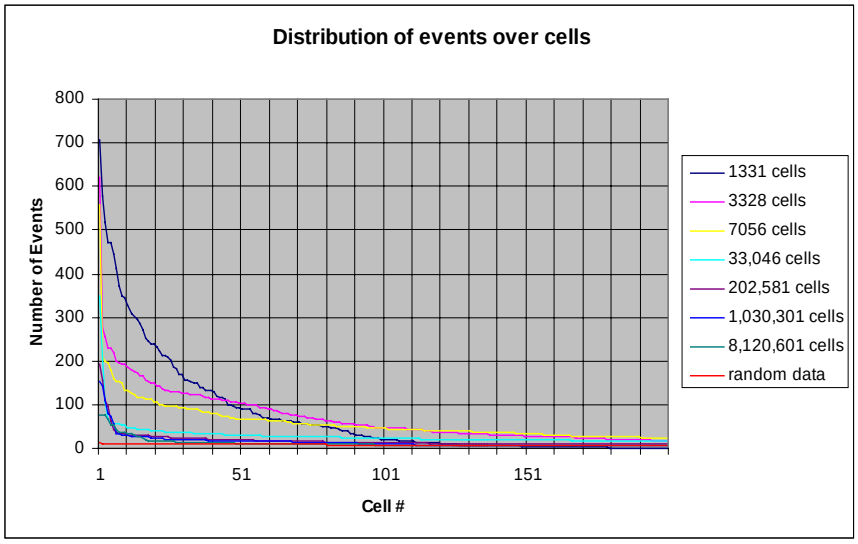
Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

- sort cells by size
- pick larger cell to start forming a cluster
- find all neighbors of “Manhattan distance” equal to 1
- include cells above a threshold
- iterate for all cells in cluster
- when no more cells above threshold, pick larger remaining cell and start forming a new cluster
- Display cluster distributions



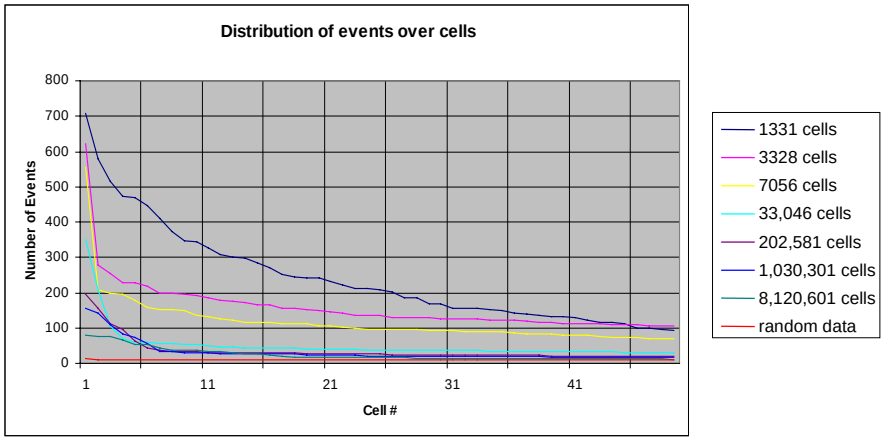
The effect of bin sizes on events distribution

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL



Blowup of cell histogram

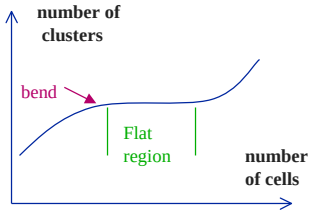
Computer Science Research and Development Department
Computing Sciences Directorate, LBNL



Cluster Stability

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

- We apply the clustering algorithm while increasing the number of bins (and thus the number of cells)
- Initially the number of clusters is small (cells are too large)
- The flat region shows stability in the number of clusters
- When cells are too small the number of cell grows
- The lower bend is the largest cell size that shows clusters

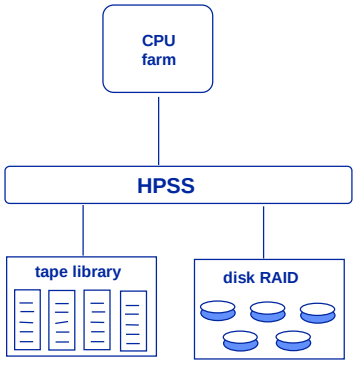


Cache Management

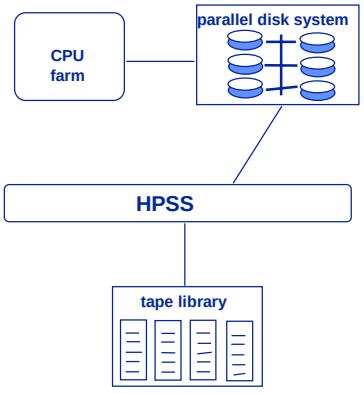
Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

Hardware scenarios

scenario A



scenario B



Cache Management Issues

- **Scenario A**
  - RAID more expensive than a Parallel Disk System (factor 2-3)
  - but, rely on HPSS to manage disk
  - storage management simplified
- **Scenario B**
  - a Parallel Disk System is cheaper, does not depend on RAID vendor
  - but, need to manage disk allocation
  - has control over placement of events on cache
- **Planned initial pilot**
  - Scenario A under NERSC

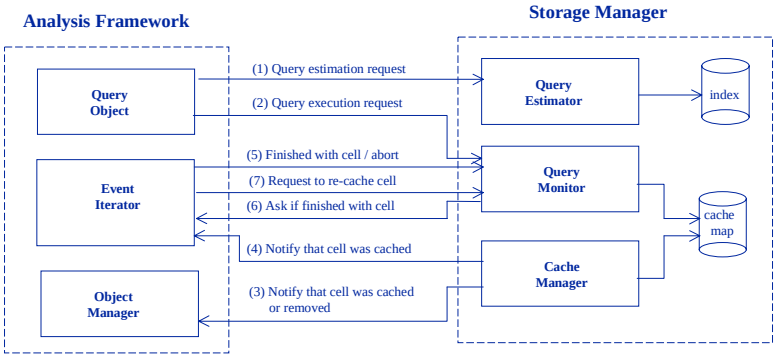
Situation at NERSC

- **To test scenario A (will be done first)**
  - tertiary storage AND cache under HPSS
  - use for “event reorganization module” only
  - dedicated experimental environment
    - rs6000 connected to HPSS, dedicated RAID partition
    - experiment with “stage”, “migrate”, and “purge”
- **To test scenario B (later)**
  - tertiary storage under HPSS, cache: DPSS
  - link not set up yet

Situation at NERSC (Con'd)

- **To test Analysis Framework communication with Storage Manager**
  - go over the net from PDSF to NT-machine using CORBA
- **To test Analysis Framework interacting with a cache (which is controlled by the Storage Manager)**
  - can't run analysis framework on T3E and C90
  - must use PDSF - DPSS

Communication between The Analysis Framework and the Storage Manager



Typical Scenario

- Query Estimation request (can be repeated many times)
- Query Execution request
- Query Monitor makes decision which cells to cache
- Cache Manager checks if cell is in cache
- If so notifies process (event iterator)
- If not it schedules a “stage cell”, “purge” unused cells
- When cells is cached, Cache Manager notifies Object Manager, as well as Event Iterator
- Event Iterator notifies Query Monitor when it is done with a cell or when it aborts

Process Misbehavior

- Use of a time out mechanism by the Query Monitor
- Query Monitor asks Event Iterator “are you alive”
- If no response within some time limit, all cells are removed for this process (if no other processes need them)
- Event Iterator can request a re-cache of a cell
- => status of queries that are “non-responsive” is saved for a period of time

Query Estimator Response

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

- **Approximate Estimate (quick response) (index in memory)**
  - No_of_events: (min,max) -- nearest bin boundaries
  - no_of_cells to be cached
  - total_MBs_to_be_moved
  - %_of_events_in_cells that qualify for a query (max)
  - no_of_events_in_cache
  - time_to_process_query
- **Precise estimate (slower response) (Index on disk)**
  - Precise No_of_events that qualify
  - ALL other measures the same

Cache Management Strategy

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

- **Number of cells to fetch ahead - parameter**
  - e.g. 2 cells => new cell cached only after process informs that it is finished with one of the 2 cells
- **Priority given to cells needed by multiple processes**
- **Priority given to cells that reside on the same tape**
- **Algorithm should not starve a process**